

## MULTI-PROCESSOR

**Publication number:** JP62163164

**Publication date:** 1987-07-18

**Inventor:** NAKAGAWA YUTAKA; SUGA RYOICHI; WATANABE YOSHIMI

**Applicant:** SONY CORP

**Classification:**

- international: **G06F15/16; G06F15/177; G06T1/00; G09G1/00; G09G5/12; H04N5/68; G06F15/16; G06T1/00; G09G1/00; G09G5/12; H04N5/68; (IPC1-7): G06F15/16; G06F15/62; G09G1/00; H04N5/68**

- European:

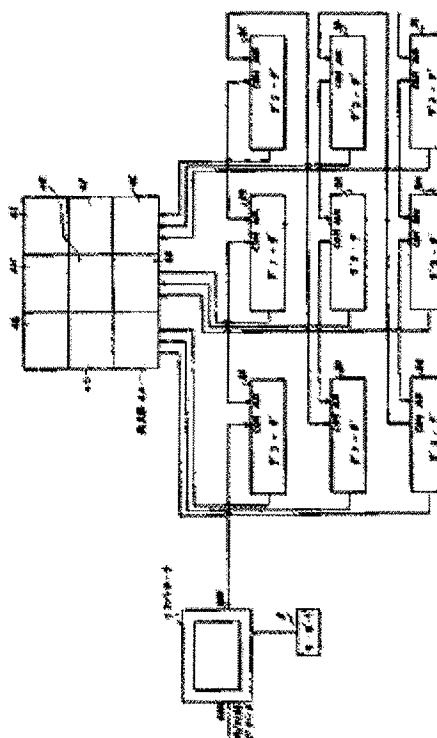
**Application number:** JP19860004763 19860113

**Priority number(s):** JP19860004763 19860113

[Report a data error here](#)

### Abstract of JP62163164

**PURPOSE:** To simplify a circuit constitution by inserting an identification number into the prescribed part of a data sequence from an information generating means, receiving the identification number through the terminal equipment of the front stage among those plural terminal equipments for initialization and producing a new identification number to transfer it to the terminal equipment of the next stage. **CONSTITUTION:** The decoders 3A-3I serving as terminal equipments are connected in series to a controller 1 serving as an information generating means, i.e., a center. Then an ID number is put into the prescribed area of the data sequence sent from the controller 1. The decoder 3A of the 1st stage receives the ID number from the controller 1 and is initialized. At the same time, the decoder 3A increases the received ID number to produce a new ID number and transfers it to the decoder 3B. Thus the decoder 3B receives the new ID number and is initialized and at the same time increases the received ID number to produce a new ID number and transfers it to the decoder 3C. Thereafter the same operations are repeated with decoders 3C-3I respectively.



Data supplied from the **esp@cenet** database - Worldwide

⑨ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭62-163164

⑤ Int.Cl.<sup>4</sup>

識別記号

庁内整理番号

⑬ 公開 昭和62年(1987)7月18日

G 06 F 15/16

M-2116-5B

15/62

6615-5B

G 09 G 1/00

7923-5C

H 04 N 5/68

C-7245-5C

審査請求 未請求 発明の数 1 (全13頁)

⑭ 発明の名称 マルチプロセッサ

⑯ 特 願 昭61-4763

⑰ 出 願 昭61(1986)1月13日

⑱ 発 明 者	中 川 裕	東京都品川区北品川6丁目7番35号	ソニー株式会社内
⑲ 発 明 者	須 賀 良 一	東京都品川区北品川6丁目7番35号	ソニー株式会社内
⑳ 発 明 者	渡 辺 好 美	東京都品川区北品川6丁目7番35号	ソニー株式会社内
㉑ 出 願 人	ソニー株式会社	東京都品川区北品川6丁目7番35号	
㉒ 代 理 人	弁理士 伊 藤 貞	外1名	

#### 明 細 書

発明の名称 マルチプロセッサ

特許請求の範囲

内部または外部からのデータに基づいて情報を発生する情報発生手段と、

該情報発生手段に対して直列の関係に配された中央処理装置を有する複数の端末器とを備え、

上記情報発生手段からのデータシーケンスの所定部に識別番号を挿入し、

上記複数の端末器のうち前段の端末器は供給された識別番号を受信して初期設定されると共に次段の端末器に対して新たな識別番号を形成して転送するようにしたことを特徴とするマルチプロセッサ。

発明の詳細な説明

以下の順序で本発明を説明する。

A 産業上の利用分野

B 発明の概要

C 従来の技術

D 発明が解決しようとする問題点

B 問題点を解決するための手段(第1図)

F 作用

G 実施例

G<sub>1</sub> 回路構成 (第1図～第3図)

G<sub>2</sub> 画面の拡大縮小表示 (第4図、第5図)

G<sub>3</sub> 拡大データ変換 (第6図、第7図)

G<sub>4</sub> ベゼル補正 (第8図、第9図)

G<sub>5</sub> ID番号の割付け (第10図、第11図)

G<sub>6</sub> 外部同期 (第12図、第13図)

G<sub>7</sub> フローコントロール (第14図、第15図)

H 発明の効果

A 産業上の利用分野

この発明は、センタに対して複数の端末器を接続し、センタより各端末器を制御する場合等に用いて好適なマルチプロセッサに関する。

B 発明の概要

この発明は、情報発生手段に複数の端末器を直列接続し、情報発生手段からのデータシーケ

スの所定部に識別番号を挿入し、複数個の端末器のうち最初の端末器は情報発生手段からの識別番号を受信すると初期設定されると共に次段の端末器に対して新たな識別番号を形成して転送し、次段の端末器は新たな識別番号を受信すると初期設定されると共に更に次段の端末器に対して新たな識別番号を形成して転送し、以下最後の端末器まで同様の動作を順次繰り返すことにより、実質的に1本の伝送ラインを用いるだけで直列接続の複数個の端末器に識別番号を割付けて初期設定を行うことができるようにしたものである。

#### C 従来の技術

或る情報を発生するセンタに対して複数個の端末器を接続し、センタより各端末器を制御する場合がある。その際にセンタと複数個の端末器との関係は一般に並列関係にある(特願昭60-23834号)。

#### D 発明が解決しようとする問題点

ところがセンタに対して複数個の端末器が並列

関係に接続されていると、各端末器毎に送信ポートが必要であると共に伝送ラインも多数必要となり、構成が複雑になると共にコスト的にも高価になる等の欠点があった。

この発明は斯る点に鑑みてなされたもので、センタに対して複数個の端末器を直列関係に接続すると共にこれ等の端末器に対して識別番号を順次割付けることができるマルチプロセッサを提供するものである。

#### E 問題点を解決するための手段

この発明によるマルチプロセッサは、内部または外部からのデータに基づいて情報を発生する情報発生手段(1)と、この情報発生手段(1)に対して直列の関係に配された中央処理装置(20)を有する複数個の端末器(3A)～(3I)とを備え、情報発生手段(1)からのデータシーケンスの所定部に識別(ID)番号を挿入し、複数個の端末器(3A)～(3I)のうち前段の端末器は供給された識別番号を受信して初期設定されると共に次段の端末器に

3

対して新たな識別番号を形成して転送するように構成している。

#### F 作用

センタである情報発生手段としてのコントローラ(1)に複数個の端末器としてのデコーダ(3A)～(3I)を直列接続する。そしてコントローラ(1)からのデータシーケンスの所定部に識別(ID)番号を挿入する。初段のデコーダ(3A)はコントローラ(1)からの識別番号を受信して初期設定されると共にその識別番号をインクリメントして新たな識別番号を形成し、デコーダ(3B)に転送する。デコーダ(3B)は新たな識別番号を受信して初期設定されると共にその識別番号をインクリメントして更に新たな識別番号を形成し、デコーダ(3C)に転送する。以下デコーダ(3C)～(3I)に付いても同様の動作が繰り返される。これにより、実質的に1本の伝送ラインを用いるだけで直列接続の複数個のデコーダに識別番号を割付けて初期設定を行うことができる。

5

4

#### G 実施例

以下、この発明の一実施例を第1図～第15図に基づいて詳しく説明する。

#### G1 回路構成

第1図は本実施例の全体の構成を示すもので、同図において、(1)は内部または外部からのデータに基づいて情報を発生する情報発生手段としてのコントローラであって、キーボード4や図示せずプリンタ等が接続されている。コントローラ(1)は通信(COMMUNICATION、以下COMと云う)ポート及び補助(AUXILIARY、以下AUXと云う)ポートを有し、COMポートに内/外部からのデータベースを受けて信号処理を行った後AUXポートより送信する。

コントローラ(1)に対して互いに直列の関係に複数個例えば9個のデコーダ(3A)～(3I)が設けられ、各デコーダ(3A)～(3I)はCOMポート、AUXポートを有する。デコーダ(3A)のCOMポートはコントローラ(1)のAUXポートと相互接

6

続され、デコーダ(3A)のAUXポートはデコーダ(3B)のCOMポートと相互接続され、デコーダ(3B)のAUXポートはデコーダ(3C)のCOMポートと相互接続され、デコーダ(3C)のAUXポートはデコーダ(3D)のCOMポートと相互接続され、以下同様に相互接続され、実質的にコントローラ(1)から最後のデコーダ(3I)まで直列関係に接続され、各COMポート及びAUXポート間で双方向伝送できるようになされている。

また、デコーダ(3A)～(3I)に対応して表示器(4A)～(4I)が設けられ、これ等の表示器

(4A)～(4I)には夫々デコーダ(3A)～(3I)の出力が供給される。つまり、ここでは $m \times n$ の画面構成を一例として $3 \times 3$ (9個)の表示器(4A)～(4I)を用いて形成する場合である。

コントローラ(1)としては例えば第2図に示すような構成のものが考えられる。すなわち同図において、(10)は中央処理装置(以下、CPUと云う)であって、このCPU(10)に対して、システムROM(11)、ワークRAM(12)、ビデオRAM(13)、カラーパレットメモリ(14)、I/Oインタフェース(15)及びフロッピディスクインタフェース(16)が設けられる。I/Oインタフェース(15)には上述のCOMポート及びAUXポートが設けられ、またこのI/Oインタフェース(15)からは後述される如く各デコーダの同期をとるための同期制御信号が発生するようになされている。

表示用のビデオRAM(13)及びカラーパレットメモリ(14)の出力側にD/A変換回路(17)が設けられる。また、CPU(10)に対してCRTコントローラ(30)が設けられ、このCRTコントローラ(30)はCPU(10)からの描画コマンドに応じてD/A変換回路(17)に描画指示を与える。そしてD/A変換回路(17)でD/A変換された信号がビデオ信号処理回路(18)で信号処理されてR、G、Bの色信号が形成され、これが表示器(19)に供給される。なおビデオRAM(13)のアドレス位置と表示器(19)のスクリーン上の画素の位置とは1対1対応とされている。

表示器(4A)～(4I)の出力側にD/A変換回路(17)が設けられる。また、CPU(10)に対してCRTコントローラ(30)が設けられ、このCRTコントローラ(30)はCPU(10)からの描画コマンドに応じてD/A変換回路(17)に描画指示を与える。そしてD/A変換回路(17)でD/A変換された信号がビデオ信号処理回路(18)で信号処理されてR、G、Bの色信号が形成され、これが表示器(19)に供給される。なおビデオRAM(13)のアドレス位置と表示器(19)のスクリーン上の画素の位置とは1対1対応とされている。

7

デコーダ(3A)～(3I)としては第3図に示すような構成のものが考えられる。すなわち同図において、(20)はCPUであって、このCPU(20)に対してシステムROM(21)、ワークRAM(22)、ビデオRAM(23)、カラーパレットメモリ(24)及びI/Oインタフェース(25)が設けられる。I/Oインタフェース(25)には上述のCOMポート及びAUXポートが設けられ、またこのI/Oインタフェース(25)には上述の同期制御信号が供給されるようになされている。

表示用のビデオRAM(23)及びカラーパレットメモリ(24)の出力側にD/A変換回路(26)が設けられる。また、CPU(20)に対してCRTコントローラ(40)が設けられ、このCRTコントローラ(40)はCPU(20)からの描画コマンドに応じてD/A変換回路(26)に描画指示を与える。そしてD/A変換回路(26)でD/A変換された信号が図示せずともビデオ信号処理回路で信号処理されてR、G、Bの色信号となり対応する

8

表示器(4A)～(4I)の一つに供給される。つまり、デコーダの構成はフロッピディスクインタフェースと表示部がない以外はコントローラ(1)と同一構成でよく、勿論I/Oインタフェース(25)に対してキーボードやプリンタ等を配するようによい。

## G. 画面の拡大縮小表示

次に画面の拡大縮小表示に付き、第4図及び第5図を参照して説明する。先ずステップ(I)でプログラムを開始し、CPU(10)によりフロッピディスクインタフェース(16)を介してディスク(図示せず)にビット列で書き込まれている或る描画コマンドを読み出してワークRAM(12)に展開し、ステップ(ロ)で描画コマンドのオペランドを解析して論理的(ユニット・スクリーン上)X-Y座標を計算し、第5図Aに示すように或る点の座標 $P(x, y)$ の値を求める。次にステップ(ハ)で求めた座標 $P(x, y)$ を $\alpha$ 倍(但し、 $\alpha$ は拡大縮小率で $\alpha \geq 0$ である)し、X

9

10

$= \alpha x$ ,  $Y = \alpha y$ より拡大縮小された結果の座標  $P'(X, Y)$  を求める。

ステップ(ニ)で求めた  $X$  の値が  $0 \leq X \leq 1$  であるかを判断し、 $0 \leq X \leq 1$  でなければ描画できないので、ステップ(ト)に進んで終了する。 $0 \leq X \leq 1$  であればステップ(ホ)に進み、ここで求めた  $Y$  の値が  $0 \leq Y \leq 1$  であるかを判断し、 $0 \leq Y \leq 1$  でなければ描画できないので、ステップ(ト)に進んで終了する。 $0 \leq Y \leq 1$  であればステップ(ヘ)に進み、筋座標情報をビデオRAM(13)の所定位置のアドレスに書き込む。このときビデオRAM(13)の所定位置のアドレス  $V\text{-RAMadd}$  は第5図Bに示すように  $V\text{-RAMadd} = \alpha y X_{\max} + \alpha x$  で決定される。つまり、第5図BはビデオRAM(13)と1対1対応の表示器(19)の表示面を表しており、 $X_{\max}$  は例えば256個の画素を表わし、 $Y_{\max}$  は200個の画素を表わしている。そして、第5図Bで  $(\alpha x, \alpha y)$  で表わされる  $P'$  点が拡大縮小された座標の描画される位置である。

11

及び(4E)により中画面を表示し、その他は単一画面とする中画面と単一画面の組み合わせや、一面大画面表示後に単一画面を入れ込むことも可能である。

また、表示器(4A)～(4I)で一面大画面を表示中に、コントローラ(10)の表示器(19)で単一画面をモニタすることも可能である。

#### G. 拡大データ変換

次に、各デコーダに対応して原データを拡大表示データに変換する場合を第5図及び第7図を参照して説明する。先ずステップ(イ)でプログラムを開始し、フロッピディスクインターフェース(16)を介してディスクにビット列で書き込まれている或る描画コマンドを読み出してワークRAM(12)に展開し、ステップ(ロ)で描画コマンドのオペランドを解析して論理的  $X-Y$  座標を計算し、 $P(x, y)$  の値を求める。

次にステップ(ハ)で  $P'(nx-i, my-j)$  により拡大された  $X-Y$  座標を求める。ただし  $n$  (横)

このようにしてビデオRAM(13)に書き込まれた拡大縮小座標情報はCRTコントローラ(30)の制御のもとに読み出され、カラーパレットメモリ(14)からの色の強さを表わす情報を付加されてD/A変換回路(17)でD/A変換されてビデオ信号処理回路(18)に供給され、ここでR、G、Bの色信号が形成され表示器(19)に表示される。

また、ビデオRAM(13)より読み出された拡大縮小情報は各デコーダに対応したID(識別)番号を付加されて、I/Oインターフェース(15)のAUXポートよりデコーダ(3A)～(3I)に供給される。各デコーダ(3A)～(3I)には自己のID番号の付加された情報を取り込んでデコードし、対応する表示器(4A)～(4I)に表示される。これにより全てのデコーダ(3A)～(3I)に与えられた情報が拡大情報であれば、表示器(4A)～(4I)の全てを用いて一面大画面が得られ、縮小情報であれば表示器(4A)～(4I)に夫々同じ単一面面が得られる。勿論その他の表示の仕方も自由であり、例えば表示器(4A)、(4B)、(4D)

12

$Xm$  (縦) 画面構成の  $(i, j)$  デコーダ用のデータである。ここで  $i, j$  は  $i = 0 \sim n-1$ ,  $j = 0 \sim m-1$  である。そして、ステップ(ニ)で  $P'(nx-i, my-j)$  を用いて描画コマンドをエンコードする。つまり拡大された  $X-Y$  座標を普通の描画コマンドに戻す。これによりデコーダ側では拡大を意識せず普通にデコードすれば結果として拡大表示が得られる。

ステップ(ホ)で全ての  $(i, j)$  について計算したか、つまり全てのデコーダに対して拡大表示データの変換が行われたかを判断し、計算してなければステップ(ヘ)に進んで  $i, j$  の値を変えて、上述同様の動作を繰り返す。そして全ての  $(i, j)$  について計算がなされた時点でステップ(ト)に進みプログラムを終了する。

因みに、 $n = 3$ ,  $m = 3$  として3倍の拡大表示のデータ変換を第7図を用いて説明する。第7図において⑥～⑩はデコーダ(3A)～(3I)に対応し、 $(i, j)$  の  $i$  を0, 1, 2,  $j$  を0, 1, 2となし、⑥のデコーダは(0, 0)、⑦のデコ

13

14

ーダは(1, 0)、②のデコーダは(2, 0)、③のデコーダは(0, 1)、④のデコーダは(1, 1)、⑤のデコーダは(2, 1)、⑥のデコーダは(0, 2)、⑦のデコーダは(1, 2)、⑧のデコーダは(2, 2)で表される。そして、 $P'(nx-1, my-j)$ を用いると、原データの座標 $P(x, y)$ は各デコーダに対して、次のように変換される。

- ①のデコーダ・・・ $P'(3x, 3y)$
- ②のデコーダ・・・ $P'(3x-1, 3y)$
- ③のデコーダ・・・ $P'(3x-2, 3y)$
- ④のデコーダ・・・ $P'(3x, 3y-1)$
- ⑤のデコーダ・・・ $P'(3x-1, 3y-1)$
- ⑥のデコーダ・・・ $P'(3x-2, 3y-1)$
- ⑦のデコーダ・・・ $P'(3x, 3y-2)$
- ⑧のデコーダ・・・ $P'(3x-1, 3y-2)$
- ⑨のデコーダ・・・ $P'(3x-2, 3y-2)$

従って、 $(x_1, y_1)$ から $(x_2, y_2)$ に向かって線を引く描画コマンドは、

①のデコーダに対して $(3x_1, 3y_1)$ から $(3x_2,$

$3y_2)$ に向かう線

②のデコーダに対して $(3x_1-1, 3y_1)$ から $(3x_2-1, 3y_2)$ に向かう線

⋮

⑨のデコーダに対して $(3x_1-2, 3y_1-2)$ から $(3x_2-2, 3y_2-2)$ に向かう線に対応する。

ここでID番号と $(i, j)$ のデコーダとの関係は $ID = jn+i$ で表される。例えば $(0, 0)$ のデコーダは0(①のデコーダ)、 $(1, 0)$ のデコーダは1(②のデコーダ)・・・ $(2, 2)$ のデコーダは8(⑨のデコーダ)の如くなる。

#### G. ベゼル補正

さて、1つの画面を複数個の表示器により表示する場合には表示器の枠(ベゼル)が問題となり、望ましくはこの枠があっても枠がないように複数個の表示器で画面を表示したい。つまり、複数個の表示器で1つの画面を表示する場合どうし

15

ても隣接する表示器の枠の所で段差が生じ、表示される画面が不自然なものとなる。そこでこれを解消する方法を次に第8図及び第9図を参照して説明する。先ず、ステップ(イ)でプログラムを開始し、フロッピディスクインターフェース(8)を介してディスクにビット列で書き込まれている或る座標を読み出してワークRAM(12)に展開し、ステップ(ロ)で描画コマンドのオペランドを解析して論理的X-Y座標を計算し、 $P'(x, y)$ の値を求める。

次にステップ(ハ)で $P'(\frac{nx-1}{\alpha}, \frac{my-j}{\alpha})$

により拡大されたX-Y座標を求める。ここで $\alpha$ は表示率で $0 \leq \alpha \leq 1$ の関係にある。たゞしこのX-Y座標は $n \times m$ 画面構成の $(i, j)$ デコーダ用のデータである。そして、ステップ(ニ)でステップ(ハ)で求めた座標を $\frac{1-\alpha}{2}$ だけ原点方向にシフトした座標 $P''(\frac{nx-1}{\alpha} - \frac{1-\alpha}{2}, \frac{my-j}{\alpha} - \frac{1-\alpha}{2})$

17

16

$\frac{1-\alpha}{2})$ を求める。

次にステップ(ホ)で全ての $(i, j)$ について計算したか、つまり全てのデコーダに対して上述の座標が求められたかを判断し、計算してなければステップ(ヘ)に進んで $i, j$ の値を変えて、上述同様の動作を繰り返す。そして、全ての $(i, j)$ について計算がなされた時点でステップ(ト)に進みプログラムを終了する。

第8図の動作に関連してデコーダ側の或る表示器の表示状態を第9図を用いて説明する。第9図において、 $a$ は表示器で表示できる物理的表示領域、 $b$ はボードとベゼル部分を含む表示器の枠である。従って、第9図では枠同士が隣接した2つの表示器を示している。第8図のステップ(ロ)でX-Y座標 $P(x, y)$ を求めると、これはデコーダ側の或る表示器の表示領域 $a$ 内に表示される。また、 $c$ は2つの表示領域 $a$ 内に描画された或る直線を表している。第8図のステップ(ハ)で拡大されたX-Y座標を求めると、これは図示

18

せずも第9図Aに直線dで示す拡大された仮想的な表示枠内に表示される。そして、この拡大された仮想的な表示枠を第9図Bに示すように原点方向に $\frac{1-\alpha}{2}$ だけシフトする。そのときの座標が第8図のステップ(ニ)で求める座標である。すると、仮想的な表示枠は第9図Bからもわかるように実際の枠bに略々一致するようになる。このとき、第9図Aで示されていた直線cは第9図Bでは少し下った位置に表示される。しかし、左側の表示領域b内の直線cと右側の表示領域b内の直線cの直線性は維持されたまゝである。つまり、隣接する表示器の枠で段差が生じることがない。

#### G: ID番号の割付け

次に各デコーダにID番号を割付ける手順を第10図及び第11図を参照して説明する。先ず、ステップ(イ)でプログラム開始し、ステップ(ロ)でデコーダ(3A)はコントローラ(1)より第10図に示すようなID割り付けのデータシーケンスが送

られているかをチェックする。ステップ(ハ)でデコーダ(3A)はコントローラ(1)より送出されてくる情報にID割り付けデータシーケンスがあるかを判断し、そうでなければステップ(ヘ)に進んでプログラムを終了し、そうであれば当該データシーケンスに含まれるID番号を自己のID番号として記憶保存する。そして初期設定される。

次にデコーダ(3A)はステップ(ホ)で自己のID番号を1つインクリメントして次段のデコーダ(3B)のID番号としてAUXポートに出力し、ステップ(ヘ)にてプログラムを終了する。

同様にデコーダ(3B)はデコーダ(3A)より供給されたID番号を自己のID番号として記憶保存し、初期設定される。そしてデコーダ(3B)は自己のID番号を1つインクリメントして次段のデコーダ(3C)のID番号としてAUXポートに出力する。以下(3D)～(3I)に付いても同様の動作が順次行われ、全てのデコーダ(3A)～(3I)に対するID番号の割り付けが終了する。

19

#### G: 外部同期

次に各デコーダに外部同期をかける場合、つまりコントローラ(1)からの同期制御信号によりデコーダ(3A)～(3I)を一斉に駆動させる場合を第12図及び第13図を参照して説明する。第12図はコントローラ(1)の動作で、第13図はデコーダ(3A)～(3I)の動作である。先ず、ステップ(イ)でプログラム開始し、ステップ(ロ)でコントローラ(1)はI/Oインターフェース(15)から出力される同期制御信号を一方のレベル例えばローレベルとする。次にステップ(ハ)でコントローラ(1)はデコーダ(3A)～(3I)に対して全てのデータを送る。ステップ(ニ)でコントローラ(1)は全てのデータ送信完了後にI/Oインターフェース(15)から出力される同期制御信号を他方のレベル例えばハイレベルにする。ステップ(ホ)でプログラムを終了する。

一方、デコーダ(3A)～(3I)は各々ステップ(イ)でプログラム開始し、ステップ(ロ)でCOMポートよりデータを受信する。ステップ

(ハ)で受信データをAUXポートに出力する。ステップ(ニ)でコントローラ(1)のI/Oインターフェース(15)より各デコーダのI/Oインターフェース(25)に供給されている同期制御信号がハイレベルか否かを判断し、ハイレベルでなければすなわちローレベルであればステップ(ロ)へ戻り、ハイレベルであればステップ(ホ)に進んでデータをデコード開始する。ステップ(ヘ)で、データ終了か否かを判断し、データ終了でなければステップ(ニ)へ戻り、データ終了であればステップ(ト)に進んでプログラムを終了する。

つまり、デコーダ(3A)～(3I)はコントローラ(1)からの同期制御信号がローレベルの間はデータを取り込むだけでデコードは行われず、同期制御信号がハイレベルになると一斉にデコード開始する。

#### G: フローコントロール

次に直列接続されたデコーダのデータのオーバーフローが検出されたら、前段のデコーダに対して

20

21

22

データ出力の停止を命令するフローコントロールの手順を第14図及び第15図を参照して説明する。まず、第14図においてコントローラ(1)はCOMポート及びAUXポートに対してワークRAM(12)上に夫々送信バッファTC及び受信バッファTRと送信バッファTA及び受信バッファRAを有しており、ここではAUXポート側の送信バッファTA及び受信バッファRAのみを示している。また、各デコードもCOMポート及びAUXポートに対してワークRAM(12)上に夫々送信バッファTC及び受信バッファRCと送信バッファTA及び受信バッファRAを有している。そして、コントローラ(1)のAUXポートの送信バッファTAのデータはデコード(3A)のCOMポートの受信バッファRCに伝送され、デコード(3A)のCOMポートの送信バッファTCのデータはコントローラ(1)のAUXポートの受信バッファRAに伝送される。つまり双方向伝送とされている。また、デコード(3A)のAUXポートの送信バッファTAのデータはデコード(3B)のCOMポートの受信

バッファRCに伝送され、デコード(3B)のCOMポートの送信バッファTCのデータはデコード(3A)のAUXポートの受信バッファRAに伝送される。つまり、この場合も双方向伝送とされている。その他のデコード間でも同様に双方向伝送できるようになされている。

このような構成において、いま、一例としてデコード(3A)～(3C)の間の動作を第15図に従って説明する。ステップ(イ)でプログラムが開始してステップ(ロ)でデコード(3B)のCOMポートの受信バッファRCがフルになったか否か、すなわち受信バッファRCがオーバフローとなったか否かが判断され、フルになるとステップ(ハ)でデコード(3B)のCOMポートの送信バッファTCに送信停止信号X offを出力する。この送信停止信号X offは前段のデコード(3A)のAUXポートの受信バッファRAで受信され、デコード(3A)はデコード(3B)へのデータの転送を停止する。ステップ(ロ)でフルになってなければステップ(ニ)に進む。

2 3

ステップ(ニ)でデコード(3B)のAUXポートの受信バッファRAがフルになったか否か判断され、フルになるとステップ(ホ)でデコード(3B)のAUXポートの送信バッファTAに送信停止信号X offを出力する。この送信停止信号X offは後段のデコード(3C)のCOMポートの受信バッファRCで受信され、デコード(3C)はデコード(3B)へのデータの転送を停止する。ステップ(ニ)でフルになってなければステップ(ヘ)に進む。

ステップ(ヘ)でデコード(3B)のCOMポートの送信バッファTCに送信停止信号X offを出力した状態か否かを判断し、出力した状態であればステップ(ト)に進む。ステップ(ト)でデコード(3B)のCOMポートの受信バッファRCに空きがあるか否かを判断し、空きがあればステップ(例)でデコード(3B)のCOMポートの送信バッファTCに送信再開信号X onを出力する。この送信再開信号X onは前段のデコード(3A)のAUXポートの受信バッファRAで受信され、デコード

2 4

(3A)はデコード(3B)へのデータの転送を再開する。ステップ(ヘ)で送信停止信号X offが出力されずまたステップ(ト)で受信バッファRCに空きがなければステップ(リ)に進む。

ステップ(リ)でデコード(3B)のAUXポートの送信バッファTAに送信停止信号X offを出力した状態か否かを判断し、出力した状態であればステップ(ヌ)に進む。ステップ(ヌ)でデコード(3B)のAUXポートの受信バッファRAに空きがあるか否かを判断し、空きがあればステップ(ル)でデコード(3B)のAUXポートの送信バッファTAに送信再開信号X onを出力する。この送信再開信号X onは後段のデコード(3C)のCOMポートの受信バッファRCで受信され、デコード(3C)はデコード(3B)へのデータの転送を再開する。そしてステップ(ヲ)でプログラムを終了する。また、ステップ(リ)で送信停止信号X offが出力されずまたステップ(ヌ)で受信バッファRAに空きがなければステップ(ヲ)に進んでプログラムを終了する。

2 5

2 6



コントローラ(1)とデコーダ(3A)及び各デコーダ間でも同様の動作が可能である。

#### H 発明の効果

上述の如くこの発明によれば、情報発生手段からのデータシーケンスの所定部に識別番号を挿入し、複数の端末器のうち前段の端末器は供給された識別番号を受信して初期設定されると共に次段の端末器に対して新たな識別番号を形成して転送するようにしたので、実質的に1本の伝送ラインを用いるだけで直列接続の複数の端末器に識別番号を割付けて初期設定を行うことができ、回路構成が簡略化される。また、情報発生手段では各端末器毎に送信ポートを設ける必要がなく、初段の端末器に対してのみ送信ポートを設ければよいので、それだけ情報発生手段の回路構成が簡略化される。

#### 図面の簡単な説明

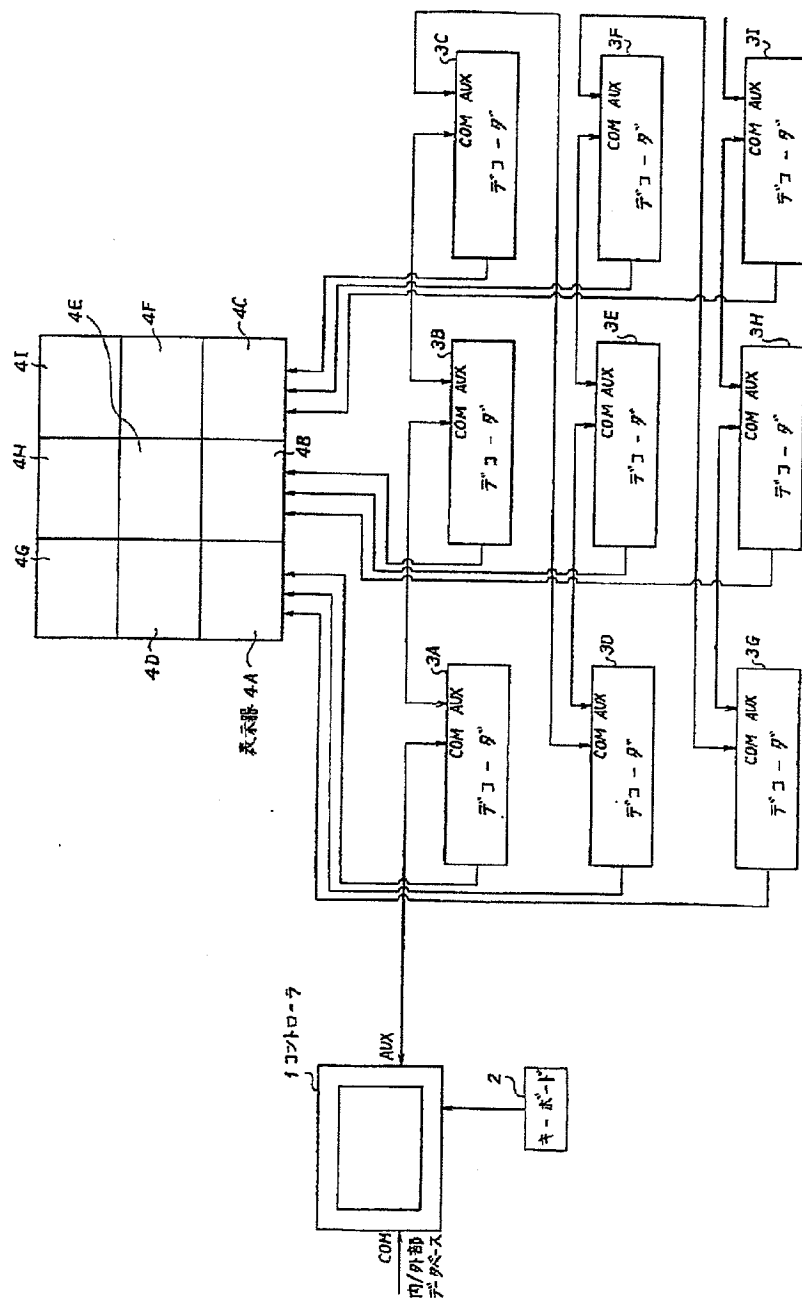
第1図はこの発明の一実施例を示す構成図、第2図は第1図で使用されるコントローラの一例を

示す構成図、第3図は第1図で使用するデコーダの一例を示す構成図、第4図及び第5図は夫々画面の拡大縮小表示の説明に供するためのフローチャート及び線図、第6図及び第7図は夫々拡大データ変換の説明に供するためのフローチャート及び線図、第8図及び第9図は夫々ベゼル補正の説明に供するためのフローチャート及び線図、第10図及び第11図は夫々ID番号の割付けの説明に供するためのフローチャート及び線図、第12図及び第13図は夫々外部同期の説明に供するためのフローチャート、第14図及び第15図は夫々フローコントロールの説明に供するための構成図及びフローチャートである。

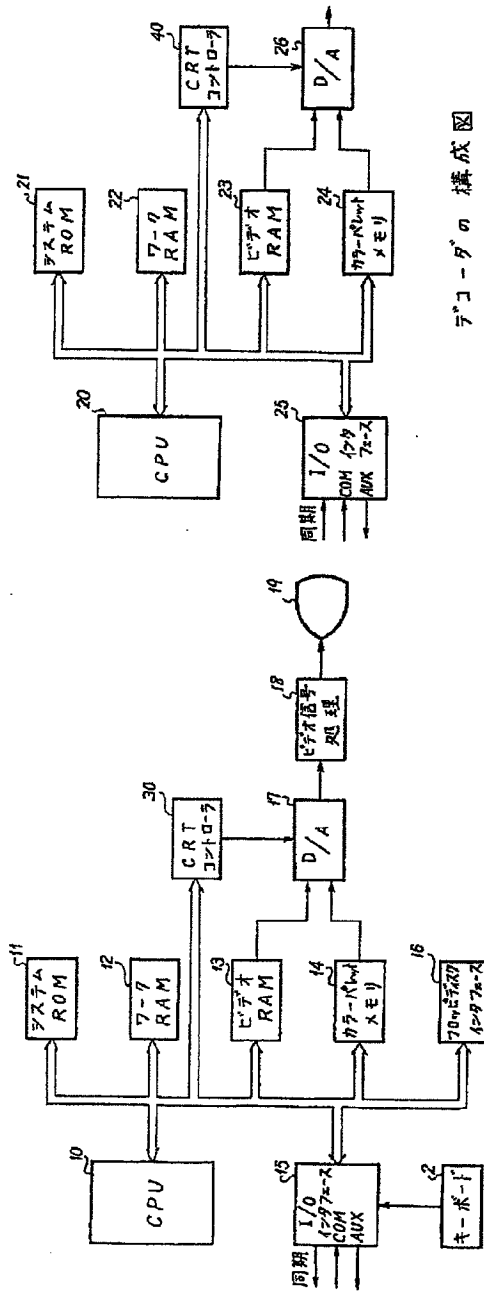
(1)はコントローラ、(2)はキーボード、(3A)～(3I)はデコーダ、(4A)～(4I)は表示器である。

代 理 人 伊 藤 貞

同 松 隈 秀 盛

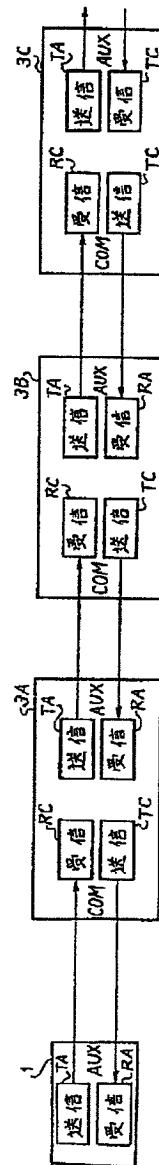


全体の構成図  
第1図

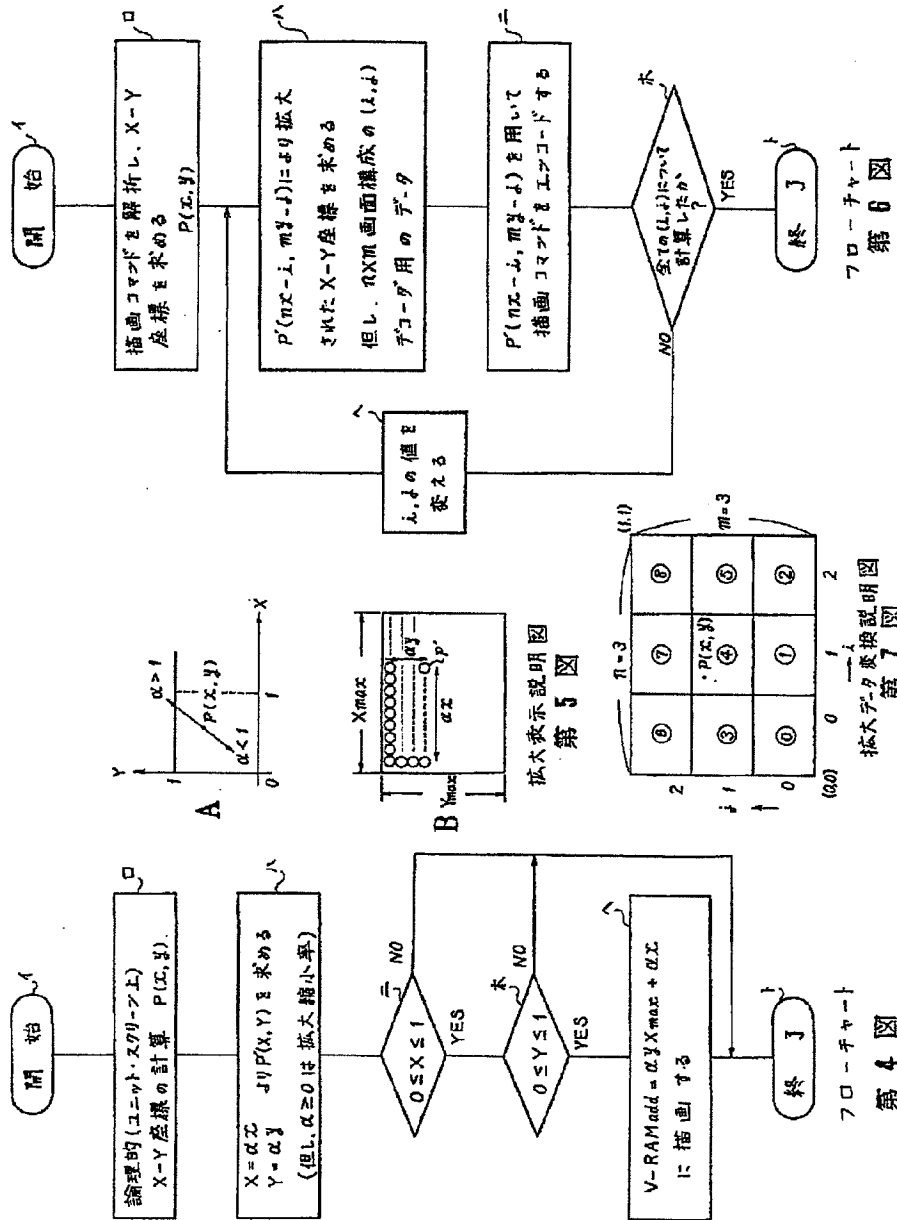


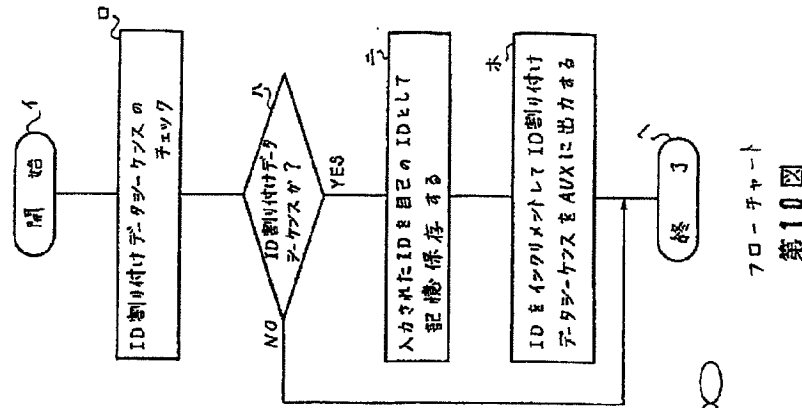
第2図  
コンピュータの構成図

第3図  
コンピュータの構成図

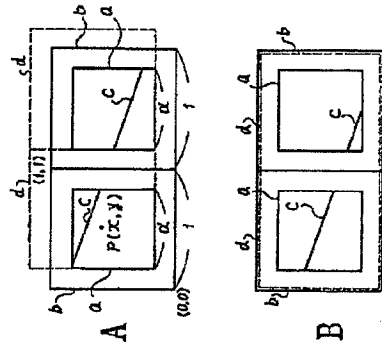


第14図  
フローコントロール説明図

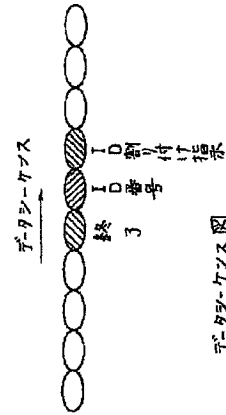




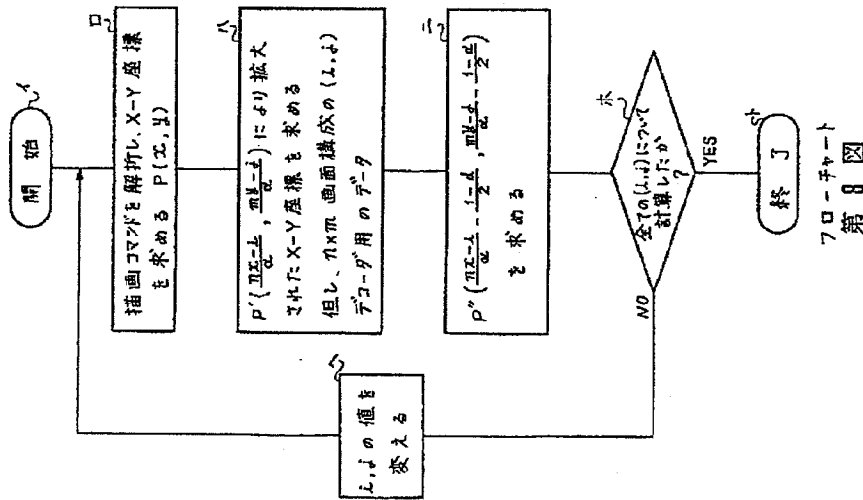
フローチャート  
第10図



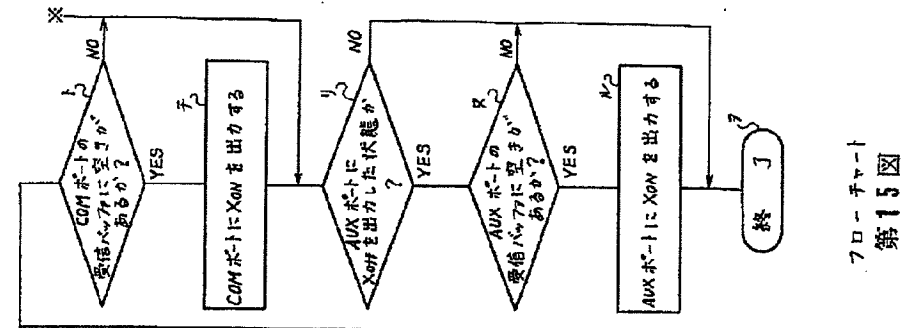
ベゼル補正説明図  
第9図



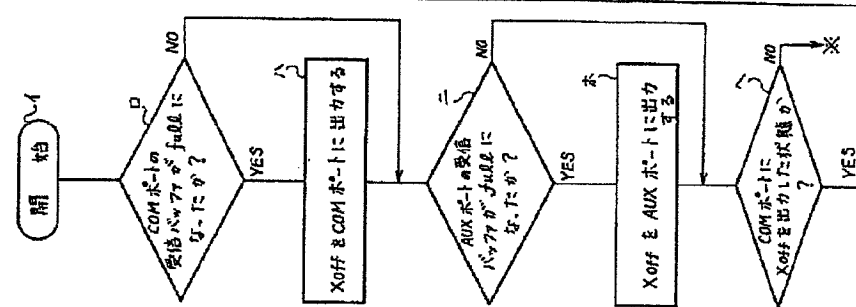
データアクセス図  
第11図



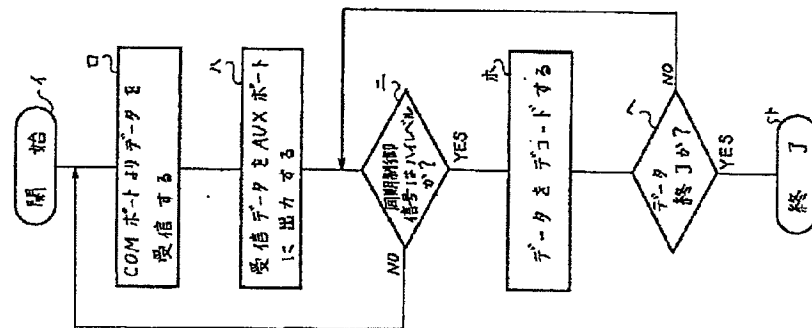
フローチャート  
第8図



フローチャート  
第15図



フローチャート  
第13図



フローチャート  
第12図

